

**IN THE CLAIMS:**

Please substitute the following claims for the claims previously presented:

1-10 (Canceled).

11. (Previously Presented) A method of supervising the execution of one or more program sections written in an object-oriented programming language to detect an object that unexpectedly disrupts execution in the one or more program sections, comprising:

(a) starting a program section and creating an object as an instance of a class;

(b) storing in a memory one or more information units associated with the created object and an expiration time period associated with the stored one or more information units, the one or more information units associated with the created object allowing supervision of execution of the program section;

(c) removing the one or more information units stored in the memory when the created object is completed;

(d) scanning the memory to identify one or more information units associated with an object that is not completed and for which there has been no activity for a time period longer than the associated expiration time period; and

(e) for the identified information unit or units in step (d), triggering an alarm signal to indicate that an unexpected disruption of execution has been detected.

12. (Previously Presented) The method according to claim 11, wherein step (b) further comprises recording a starting time of the expiration time period.

13. (Previously Presented) The method according to claim 11, further comprising:  
designating the uncompleted object for which there has been no activity for a time period longer than the associated expiration time period as being in a lost state.

14. (Previously Presented) The method according to claim 11,  
wherein the alarm signal indicates that a failure associated with the lost object has been  
detected.

15. (Canceled).

16. (Canceled).

17. (Canceled).

18. (Previously Presented) The method according to claim 11, further comprising:  
maintaining statistical information about a number of instances in each class.

19. (Previously Presented) The method according to claim 18, further comprising:  
generating a message when a usage volume exceeds a predetermined level.

20. (Previously Presented) A method of supervising the execution of one or more  
program sections written in an object-oriented programming language to detect an object that  
unexpectedly disrupts execution in the one or more program sections, comprising:

(a) starting a program section and creating an object;

(b) storing one or more information units identifying the created object in a memory, the  
one or more information units associated with the created object allowing supervision of  
execution of the program section;

(c) removing one or more information units stored in the memory when the created  
object is completed;

(d) scanning the memory to identify one or more information units associated with an  
object that is not completed and for which there has been no activity for a time period longer  
than a predetermined time period; and

(e) sending an alarm signal for each information unit identified in step (d) when the created object is inactive to indicate that an unexpected disruption of execution has been detected.

21. (Previously Presented) The method in claim 20, further comprising:

(f) delaying a transmission of the alarm signal for each information unit identified in step (d) when the created object is active.

22. (Previously Presented) Apparatus for supervising the execution of one or more program sections written in an object-oriented programming language to detect an object that unexpectedly disrupts execution in the one or more program sections, comprising:

electronic circuitry configured to start a program section and creating an object as an instance of a class; and

a memory for storing one or more information units associated with the created object and an expiration time period associated with the stored one or more information units, the one or more information units associated with the created object allowing supervision of execution of the program section;

the electronic circuitry further configured to:

remove the one or more information units stored in the memory unit when the created object is completed;

scan the memory to identify one or more information units associated with an object that is not completed and for which there has been no activity for a time period longer than the associated expiration time period; and

trigger an alarm signal in response to the identified information unit or units to indicate that an unexpected disruption of execution has been detected.

23. (Previously Presented) The apparatus according to claim 22, wherein the electronic circuitry is further configured to record a starting time of the expiration time period.

24. (Previously Presented) The apparatus according to claim 22, wherein the electronic circuitry is further configured to detect the uncompleted object for which there has been no activity for a time period longer than the associated expiration time period as being in a lost state.

25. (Previously Presented) The apparatus according to claim 24, wherein the electronic circuitry is further configured to delay termination of the program section after lapse of the expiration time period if the created object is active.

26. (Previously Presented) The apparatus according to claim 24, wherein the alarm signal indicates that a failure associated with the lost object has been detected.

27. (Canceled).

28. (Canceled).

29. (Previously Presented) The apparatus according to claim 22, wherein the electronic circuitry is further configured to:

maintain statistical information about a number of instances in each class.

30. (Previously Presented) The apparatus according to claim 27, wherein the electronic circuitry is further configured to:

generate a message when a usage volume exceeds a predetermined level.

31. (Previously Presented) An apparatus for supervising the execution of one or more program sections written in an object-oriented programming language, comprising:

electronic circuitry configured to start a program section and creating an object as an instance of a class; and

a memory for storing one or more information units identifying the created object;

the electronic circuitry further configured to:

terminate the program section;

remove one or more information units stored in the memory when the created object is completed or inactive;

scan the memory to identify one or more information units having been stored in the memory for a time period longer than a predetermined time period; and

send an alarm signal in response to the identified information unit or units.

32. (Previously Presented) The apparatus in claim 31, wherein the electronic circuitry is further configured to send the alarm signal only when the one or more information units have been stored in the memory for a time period longer than a predetermined time period and the created object is inactive.

33. (Previously Presented) The apparatus in claim 31, wherein the electronic circuitry is further configured to delay sending the alarm signal when the created object is active.

34. (New) A method of automatically detecting one or more objects that unexpectedly disrupt execution in an object-oriented software application, comprising;

(a) registering an object in a persistent location;

(b) specifying an expected expiration time associated with the object indicating an expected expiration time when the object is expected to terminate;

(c) deregistering the object, when the object terminates by the expected expiration time, by removing the object from the persistent location;

(d) supervising the persistent location to detect an object that has not been deregistered after expiration of the expected expiration time associated with the object;

(e) identifying the detected object; and

(f) creating a report indicating an operational failure for the identified object.

35. (New) The method according to claim 34, wherein step (d) further comprises testing the detected object to determine whether the object is active or not active, and if the object is active, resetting the expected expiration time associated with the object.